PDP-1 COMPUTER
ELECTRICAL ENGINEERING DEPARTMENT
M.I.T.
CAMBRIDGE, MASSACHUSETTS 02139

PDP-35

PDP-1 INSTRUCTION LIST

August 7, 1968

PDP-1 INSTRUCTION LIST

This list includes the title of the instruction, the normal execution time** (i.e., the time with no indirect address*), the mnemonic code of the instruction, the operation code number, and the description of the instruction's operation. In the following list, the Accumulator is abbreviated as AC, the In-Out Register as IO, and the contents of a register as C(). Thus, C(Y) indicates the contents of memory at Address Y, C(AC), the contents of the Accumulator: and C(IO), the contents of the In-Out Register.

I. MEMORY REFERENCE INSTRUCTIONS

A. Information Transfer

1. Load Accumulator (10 μsec)

   lac Y Operation Code 20

   The C(Y) are placed in the Accumulator. The C(Y) are unchanged. The original C(AC) are lost.

2. Deposit Accumulator (10 μsec)

   dac Y Operation Code 24

   The C(AC) replace the C(Y) in the memory. The C(AC) are left unchanged by this instruction. The original C(Y) are lost.

3. Deposit Address Part (10 μsec)

   dap Y Operation Code 26

   Bits 6 through 17 of the accumulator replace the corresponding digits of memory register Y. The C(AC) are unchanged as are the contents of bits 0 through 5 of Y. The original contents of bits 6 through 17 of Y are lost.

* Add 5 microseconds for each indirect address level.
** See additional notes.

4. Deposit Instruction Part (10 μsec)

   dip Y Operation Code 30

   Bits 0 through 5 of the Accumulator replace the corresponding digits of memory register Y. The C(AC) are unchanged as are the bits 6 through 17 of Y. The original contents of bits 0 through 5 of Y are lost.

5. Load In-Out Register (10 μsec)

   lio Y Operation Code 22

   The C(Y) are placed in the In-Out Register. The C(Y) are unchanged. The original C(IO) are lost.

6. Deposit In-Out Register (10 μsec)

   dio Y Operation Code 32

   The C(IO) replace the C(Y) in memory. The C(IO) are unaffected by this instruction. The original C(Y) are lost.

7. Deposit Zero in Memory (10 μsec)

   dzm Y Operation Code 34

   Clears (sets equal to plus zero) the C(Y).

8. Load Index Register (10 μsec)

   lxr Y Operation Code 12

   The C(Y) are placed in the Index Register.
   The C(Y) are unchanged. The original C(XR) are lost.

B. Arithmetic Instructions

1. Add (10 μsec)

add Y Operation Code 40

The sum of the C(Y) and the C(AC) replace the C(AC). The C(Y) are unchanged. The addition is performed with one's complement arithmetic. If the sum of two like-signed numbers yields a result of the opposite sign, the overflow flip-flop will be set (see szo instruction). A result of minus zero is changed to plus zero. Note that (-0)+(-0)=+0 and overflow is turned on.

2. Add to Memory (10 μsec)

adm Y Operation Code 36

The sum of the C(Y) and the C(AC) replace both the C(Y) and the C(AC). Otherwise, it is similar to the add instruction.

3. Subtraction (10 μsec)

sub Y Operation Code 42

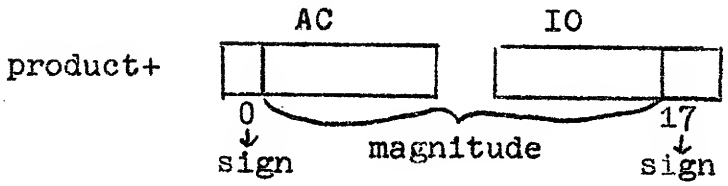
The C(AC) minus the C(Y) replace the C(AC). The C(Y) are unchanged. The subtraction is performed using one's complement arithmetic. When two unlike-signed numbers are subtracted, the sign of the result must agree with the sign of the original Accumulator, or the overflow flip-flop will be set (see szo instruction). A result of minus zero can exist in one instance only. (-0)-(+0)=(-0).

4. Multiply (14 to 25 μSec)

mul Y operation Code 54

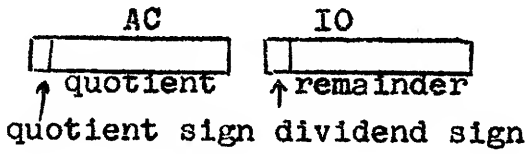
The product of the C(AC) and the C(Y) is formed in the AC and IO registers. The sign of the produce is in the AC sign bit. IO bit 17 also contains the sign of the product. The magnitude of the product is the 34-bit string from AC Bit 1 through IO bit 16. The C(Y) are not affected by this instruction. If the entire product results in a minus zero, it is changed

to a plus zero. The variation in execution time is caused by the number of one's in C(Y)



5. Divide (30 to 40 μsec, except on overflow, 12 μsec)
div Y Operation Code 56

The dividend must be in the AC and IO registers in the form indicated in the instruction, mul. IO bit 17 is ignored. The divisor is the C(Y). At the quotient and the C(IO) are the remainder. The sign of the remainder (in IO bit zero) is the sign of the dividend. The instruction that follows a div instruction will be skipped unless an overflow occurs. The C(Y) are not affected by this instruction. If the remainder of quotient result in minus zero, that value is changed to plus zero. If the magnitude of the high order part of the dividend is equal to or greater than the magnitude of the divisor, an overflow is indicated. In this case, the instruction following the div is not skipped. The original C(AC) and C(IO) are restored The overflow flip-flop is not affected.



6. Index (10 μsec)
idx Y Operation Code 44

C(Y) + 1 replace the C(Y) and the C(AC). The previous C(AC) are lost. Overflow is not indicated. If the original C(Y) equals the integer, minus one (-1), the result after indexing is plus zero.

C. Logical Instructions

1. Logical AND (10 μsec)

and Y Operation Code 02

The bits of C(Y) operate on the corresponding bits of the Accumulator to form the logical AND. The result is left in the Accumulator. The C(Y) are unaffected by this instruction.

Logical AND Table

AC bit	Y bit	Result
0	0	0
0	1	0
1	0	0
1	1	1

2. Exclusive OR (10 μsec)

xor Y Operation Code 06

The bits of C(Y) operate on the corresponding bits of the Accumulator to form the exclusive OR. The result is left in the Accumulator. The C(Y) are unaffected by this instruction.

Exclusive OR table

AC bit	Y bit	Result
0	0	0
0	1	1
1	0	1
1	1	0

3. Inclusive OR (10 μsec)

ior Y Operation Code 04

The bits of C(Y) operate on the corresponding bits of the Accumulator to form the inclusive OR. The result is left in the Accumulator. The C(AC) are unaffected by this instruction.

Inclusive OR table

AC bit	Y bit	Result
0	0	0
0	1	1
1	0	1
1	1	1

D. Decision-Making Instructions

1. Skip if Accumulator and Y Differ (10 μsec)

sad Y Operation Code 50

The C(Y) are compared with C(AC). If the two numbers are different, the Program Counter is indexed one extra position and the next instruction in the sequence is skipped. The C(AC) and the C(Y) are unaffected by this operation.

2. Skip if Accumulator and Y are the Same (10 μsec)

sas Y Operation Code 52

The C(Y) are compared with C(AC). If the two numbers are identical, the Program Counter is indexed one extra position and the next instruction in the sequence is skipped. The C(AC) and the C(Y) are unaffected by this operation.

3. Index and Skip if Positive (10 μsec)

isp Y Operation Code 46

The C(Y) +1 replace the C(Y) and the C(AC). The previous C(AC) are lost. If, after the addition the Accumulator is positive, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped. Overflow is not indicated. If the original C(Y) equals the integer, minus one (-1), the result after indexing is plus zero and the skip takes place.

E. Transfer Instructions

1. Jump (5 μsec)

jmp Y Operation Code 60

The next instruction executed will be taken from Memory Register Y. The Program Counter is reset to Memory Address Y. The original contents of the Program Counter are lost.

2. Jump and Save Program Counter (5 μ sec)

jsp Y Operation Code 62

The contents of the Program Counter are transferred to bits 6 through 17 of the Accumulator. The state of the overflow flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program Counter to bits 3, 4, and 5 of the AC. When the Transfer of the PC to the AC takes place, the Program Counter holds the address of the instruction following the jsp. The Program Counter is then reset to Address Y. The next instruction executed will be taken from Memory Register Y. The original C(AC) are lost.

3. Jump and Deposit Accumulator (10 μsec)

jda Y Operation Code 17

The contents of the AC are deposited in Memory Register Y. The contents of the Program Counter (holding the address of the instruction following the jda instruction) are transferred to bits 6 through 17 of the AC. The state of the overflow flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program Counter to bits 3, 4, and 5 of the AC. The next instruction executed is taken from Memory Register Y+1. The jda instruction requires that the indirect bit be a one, but indirect addressing does not occur. The instruction is equivalent to the instruction dac Y followed by jsp Y+1.

4. Jump and Deposit Program Counter (10 μsec)
jdp Y Operation Code 14

The contents of the Program Counter (holding the address of the instruction following the jdp) are deposited in bits 6 through 17 of the Memory Register Y. The original contents of the AC remain in the AC unchanged. The state of the overflow flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program Counter to bits 3, 4, and 5 of the Memory Register Y. The next instruction executed is taken from Memory Register Y+1.

5. Call Subroutine (10 μsec)
cal Y Operation Code 16

The address part of the instruction, Y, is ignored. The contents of the AC are deposited in the Memory Register 100 of core 0. The contents of the Program Counter (holding the address of the instruction following the cal) are transferred to bits 6 through 17 of the AC. The state of the overflop flip-flop is transferred to bit zero, the condition of the Extend flip-flop to bit 1, and the contents of the Extended Program counter to bits 3, 4, and 5 of the AC. The next instruction executed is taken from Memory Register 101. (See above.) The cal instruction requires that the indirect bit be zero. The instruction may be used as part of a master routine to call subroutines.

II Augmented Instructions

1. Load Accumulator with N (5 sec)

law N Operation Code 70

The number in the memory address bits of the instruction word is placed in the Accumulator. If the indirect address bit (bit 5) is a one (1), (-N) is put in the Accumulator.

A. Shift Group (5 sec)

sft Operation Code 66

This group of instructions will rotate or shift the Accumulator and/or the In-Out Register. When the two registers operate combined, the In-Out Register is considered to be an 18-bit magnitude extension of the right end of the Accumulator.

a. Rotate is a non-arithmetic cyclic shift. That is, the two ends of the register are logically tied together and information is rotated as though the register were a ring.

b. Shift is an arithmetic operation and is, in effect, multiplication of the number in the register by $2^{\pm N}$, where N is the number of shifts: plus is left and minus is right. As bits are shifted out from one end of a register they are replaced at the other end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted. The number of shift or rotate steps to be performed (N) is indicated by the number of one's (1's) in bits 9 through 17 of the instruction word. Thus, Rotate Accumulator Right nine times is 671777. A shift or rotate of one place can be indicated nine different ways. The usual convention is to use the right end of the instruction word. (rar 1 =671001)

1. Rotate Accumulator Right (5 μsec)
   rar N Operation Code 671
   Rotates the bits of the Accumulator right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.
2. Rotate Accumulator Left (5 μsec)
   ral N Operation Code 661
   Rotates the bits of the Accumulator left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.
3. Shift Accumulator Right (5μsec)
   sar N Operation Code 675
   Shifts the contents of the Accumulator right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the right end of the AC they are replaced at the left end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.
4. Shift Accumulator Left (5 μsec)
   sal N Operation Code 665
   Shifts the contents of the Accumulator left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the left end of the AC they are replaced at the right end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.
5. Rotate In-Out Register Right (5μsec)
   rir N Operation Code 672
   Rotates the bits of the In-Out Register right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

6. Rotate In-Out Register Left (5 μsec)
ril N Operation Code 662

Rotates the bits of the In-Out Register left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

7. Shift In-Out Register Right (5 μsec)
sir N Operation Code 676

Shifts the contents of the In-Out Register right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the right end of the IO they are replaced at the left end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

8. Shift In-Out Register Left (5 μsec)
sil N Operation Code 666

Shifts the contents of the In-Out Register left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the left end of the IO they are replaced at the right end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

9. Rotate Accumulator and In-Out Right (5 μsec)
rcr N Operation Code 673

Rotates the bits of the combined registers right in a single ring N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

10. Rotate Accumulator and In-Out Left (5 μsec)
rcl N Operation Code 663

Rotates the bits of the combined register left in a single ring N position, where N is the number of one's (1's) in bits 9 through 17 of the instruction word.

11. Shift Accumulator and In-Out Right (5 μsec)

scr N Operation Code 677

Shifts the contents of the combined register right N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the right end of the IO they are replaced at the left end of the AC by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

12. Shift Accumulator and In-Out Left (5 μsec)

scl N Operation Code 667

Shifts the contents of the combined registers left N positions, where N is the number of one's (1's) in bits 9 through 17 of the instruction word. As bits are shifted out from the left end of the AC they are replaced at the right end of the IO by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

F. Miscellaneous

1. Execute (5 μsec plus time of instruction executed)
   xct Y Operation Code 10

   The instruction located in register Y is executed. The Program Counter remains unchanged (unless a jump or skip were executed). If a skip instruction is executed (by xct Y), depending on the skip condition either the Program Counter is advanced one extra position and the next instruction in the sequence (after the xct instruction) is skipped or the Program Counter remains unchanged and the next instruction in the sequence is executed. Execute may be indirectly addressed, and the instruction being executed may use indirect addressing. An xct instruction may execute other xct commands.

B. Skip Group (5 μsec)

skp Operation Code 64

This group of instructions senses the state of various flip-flops and switches in the machine. The address portion of the instruction selects the particular function to be sensed. All members of this group have the same operation code The in tructions in the Skip Group may be combined to form the inclusive OR of the separate skips. Thus, if Address 3000 is selected, the skip would occur if the overflow flip-flop equals 0 (zero) or if the In-Out Register is positive.

a. The combined instruction would still take 5 microseconds.

b. The intents of any skip instruction can be reversed by making bit 5 (normally the indirect address bit) equal to one (1). For example, the skip on Zero Accumulator instruction, with bit 5 equal to one (1), becomes Do Not Skip on Zero Accumulator.

1. Skip on Zero Accumulator (5 μsec)

sza Address 0100

If the Accumulator is equal to plus zero (all bits zre zero), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

2. Skip on Plus Accumulator (5 μsec)

spa Address 0200

If the sign bit of the Accumulator is zero (0), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

3. Skip on Minus Accumulator (5 μsec)

sma Address 0400

If the sign bit of the Accumulator is a one (1)l the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

4. Skip on Zero Overflow (5 sec)

szo Address 1000

If the overflow flip-flop is a zero (0), the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. The overflow flip-flop is cleared by the instruct on. This flip-flop is set only by an addition or subtraction that exceeds the capacity of the accumulator. (see definition of add and subtract instructions.) The overflow flip-flop is not cleared by arithmetic operations which do not cause an overflow. Thus, a whole series of arithmetic operations can be checked for correctness by a single szo. The overflow flip-flop is also cleared by the "start" switch.

5. Skip on Plus In-Out Register (5 sec)

spi Address 2000

If the sign bit of the In-Out register is zero, the Program Counter is indexed one extra position, and the next instruction in sequence is skipped.

6. Skip on Non-Zero In-Out Register (5 sec)

sni Address 4000

If the In-Out Register is not equal to plus zero (at least one bit is a one), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

7. Skip on Zero Switch (5 sec)

szs Address 0010, 0020, 0030, ... 0070

If the selected Sense Switch is zero the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. Address 10 senses the position of Sense Switch 1, Address 0020, Switch 2, etc. Address 0070 senses all the switches. If Address 0070 is selected all 6 switches must be zero (0) to cause the skip. The instruction to skip on zero senee switch 1 would be szs 10, for sense switch 2, szs 20, etc.

8. Skip on Zero Program Flag (5 μsec)

szf Addresses 0001 to 0007

If the selected program flag is a zero, the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. Address 0001 selects Program Flag 1, etc. Address 0007 selects all six program flags. All six must be zero to cause the skip. The instruction to skip on zero program flag 1 would be szf 1, for program flag 2, szf 2, etc.

The following skip group instructions are extended instructions: They are defined to represent the or-ing of existing skip instructions.

9. Skip on Zero or Minus Accumulator (5 μsec)

szm

If the sign bit of the Accumulator is a one (1) or if the Accumulator is equal to plus zero (all bits are zero), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped. This represents the or-ing of sza and sma.

10. Skip on Positive Quantity (5 μsec)

spq

If the Accumulator is not equal to plus zero (all bits are zero) and if the sign bit of the accumulator is not a one (1), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped. This represents the or-ing of sza i and sma i.

11. Clear the Overflow Flip-flop (5 μsec)

clo

The overflow flip-flop is set only by addition or subtraction that exceeds the capacity of the AC. The overflow flip-flop is cleared by the instruction szo: it is not cleared by any arithmetic operations. This instruction clears the overflow flip-flop and then proceeds to the next instruction in the sequence. This represents the or-ing of szo, sma, spa, and i.

C. Operate Group (5 μsec)
opr Operation Code 76

This instruction group performs miscellaneous operations on various Central Processor Registers. The address portion of the instruction specifies the action to be performed. The instructions in the Operate Group can be combined to give the union of the functions. For example, the instruction opr 3200 will clear the AC, put the contents of the Text Word into the AC, and complement the AC.

a. The combined instruction would still take 5 microseconds.

b. The order of operate class instructions is:

EFFECT	0→AC 0→IO	TW→AC, stf clf	$\overline{AC}$→AC 0→RUN $\overline{IO}$→IO	0→MB (lai)	IO→MB(lai) C→IO(lia)	AC→MB(lia) MB→AC(lai)	MB→IO (lia)
	7	8	9	10	0	1	2
TIME	Time pulses of this instruction				TPs of next instruction		

1. Clear the In-Out Register (5 μsec)
cli Address 4000
Clears (set equal to plus zero) the In-Out Register.

2. Load the Accumulator from the Test Word (5 μsec)
lat Address 2000
Forms the inclusive OR of the C(AC) and the contents of the Test Word. This instruction is usually combined with Address 0200 (Clear Accumulator), so that C(AC) will equal the contents of the Test Word Switches. (Thus, lat is defined as 762200 initially in certainly symbol table and ID symbol table.)

3. Complement the In-Out Register (5 μ sec)
   cmi Address 0100
   Complements (changes all ones to zeroes and all zeroes to ones) the contents of the In-Out Register.

4. Complement the Accumulator (5 μ sec)
   cma Address 1000
   Complements (changes all ones to zeroes and all zeroes to ones) the contents of the Accumulator.

5. Halt
   hlt Address 0400
   Stops the computer. When the computer is in Time-Sharing, this instruction is treated as an illegal instruction when executed.

6. Clear the Accumulator (5 μ sec)
   cla Address 0200
   Clears (sets equal to plus zero) the contents of the Accumulator.

7. Clear Selected Program Flag (5 μ sec)
   clf Address 0001 to 0007
   Clears the selected Program Flag. Address 0001 clears Program Flag 1, 0200 clears Program Flag 2, etc. Address 0007 clears all program flags. Thus, the instruction to clear Program Flag 1 is clf 1, for Program Flag 2, clf 2, etc.

8. Set Selected Program Flag (5 μ sec)
   stf Address 0011 to 0017
   Sets the select porgram flag. Address 0011 sets Program Flag 1: 0012 sets Program Flag 2, etc. Address 0017 sets all program flags. Thus, if stf is defined as 760010, the instruction to set Program Flag 1 is stf 1, for Program Flag 2, stf 2, etc. (Stf is initially defined in CERTAINLY symbol table and ID symbol table as 760010.)

9. No Operation (5 μsec)

nop Address 0000

The state of the computer is unaffected by this operation, and the Program Counter continues in sequence.

10 10. Load the Accumulator from the In-Out Register (5 sec)

lai Address 0040

This instruction copies the contents of the in-out register into the Accumulator. It happens after all normal operate class options. If the computer is stopped at the end of this instruction, the Memory Buffer Register will contain zero, and the old contents of the Accumulator will be shown. (See the order of implementation of the operation class instructions at the beginning of this section.)

11. Load the In-Out Register from the Accumulator (5 sec)

lia Address 0020

This instruction copies the contents of the Accumulator into the in-out register. It happens after all normal operate class options. If the computer is stopped at the end of this instruction, the Memory Buffer Register will contain zero, and the old contents of the In-Out Register will be shown. (See the order of implementation of the operate class instructions at the beginning of this section.)

The following operate group instructions are extended instructions; they are defined to represent the or-ing of existing operate instructions

12. Swap Accumulator with the In-Out Register (5 μ sec)
swp

This instruction is the combination of lia and lai. It copies the contents of the Accumulator into the In-Out Register and the original contents of the In-Out Register is copied into the Accumulator. It happens after all normal operate class options. If the computer is stopped at the end of this instruction the Memory Buffer Register will contain zero, and the swap will not yet have occurred.

13. Clear and Complement the Accumulator (5 μ sec)
clc

This instruction represents the or-ing of cla and cma. It clears (sets equal to plus zero) the contents of the Accumulator and then complements (changes all ones to zeroes and all zeroes to ones) the contents of the Accumulator.

LII. IN-OUT TRANSFER GROUP (5 μsec without in-out wait)

iot Operation Code 72

The in-out transfer command is used to perform all the in-out control and information transfer functions. The address of the iot instructions is used to select various devices. The decoding for the instructions is as follows:

Bits	Use
0-4	11101 instruction bit code for in-out transfer
5	Used for device synchronization
7-11	Useful for modification of iot instructions
12-17	Addresses 1 of 64 possible device

NOTE: Quite often, input-output operations must be synchronized. That is, information is transferred which may cause the computer (or device) to "wait", and then proceed in synchronism. When several in-out devices operate simultaneously, the synchronization is essential. The control for this is coded in each in-out transfer command.

IN-OUT TRANSFER Command Bits	Wait for Completion Pulse for Restart/Continue	Enable/Disable Completion (Done) Pulse Signal
5	without wait	
0	continue, no wait	Disable
1	wait, then continue	Enable

Bit 5 of the in-out transfer command designates whether the program is to wait for a completion pulse before continuing and whether the completion pulse return signal is to be enabled or disabled.

The description of each in-out instruction below is divided into two parts, NON-TS and TS, so that the user can understand the differences. Generally, in time sharing mode the executive routine buffers characters for the typewriter, punch, and reader so the user does not worry about synchronization. Almost all iot instructions trap to the executive routine where they are interpreted and serviced. Control is quickly returned to the user's program except when an output instruction would cause an overflow of an executive routine buffer. In this case, the program is dismissed and remains inactive until the buffer becomes almost empty.

A. Reader

The perforated tape reader of the PDP-1 is a photoelectric device capable of reading 320 or 640 lines per second. Three lines form the standard 18-bit word when reading binary punched 8-hole tape. Five, six, and seven-hole tape may also be read.

1. Read Paper Tape, Alphanumeric
   rpa Address 0001
   a. NON-TS: This instruction reads one line of tape (all eight channels) and transfers the resulting 8-bit code to the Reader Buffer. (NOTE: rpa is initially defined as 730001 in ID symbol table and certainly symbol table.
      1. If bit 5 of the rpa instruction is a "1" [730001≡(rpa)] the 8-bit code read from tape is automatically transferred to the IO register via the Reader Buffer and appears as follows:

IO BITS	10	11	12	13	14	15	16	17
TAPE CHANNELS	8	7	6	5	4	3	2	1

      The remaining bits of the IO are set to zero. The program waits until the material has been read and transferred and a completion pulse returned before continuing.

2. If bit 5 of the rpa instruction is a zero [(rpa-i)=720001], the contents of the Reader Buffer must be transferred to the IO Register by executing a rrb instruction. When the Reader Buffer has information ready to be transferred to the IO Register, Status Register bit 1 is set to one. (Sequence break can be used in this situation to perform the read.)

b. TS: The reader must be assigned to the user's console before his program can execute a rpa instruction. (When the reader is not assigned, the rpa instruction is an illegal one.) The instruction is the same as in NON-TS but the the line read is put into a pseudo reader buffer (prb). The executive reader buffer routine is then filled with succeeding lines of tape. The contents of the prb is then placed in the IO if the instruction was a rpa. Otherwise, on rpa-i, the material goes into the prb but isn't transferred to the IO until a rrb instruction is executed by the user. Material is taken from the reader buffer in the executive routine on the next rpa instructions. When the buffer is almost empty more tape is read in to refill it. NOTE: The code of the off-line tape preparation typewriter (Friden FIO-DEC Recorder - Reproducer) contains an odd parity bit. This bit may bit may be checked be read-in programs. The FIO-DEC code can then be converted to the Concise (6 bit) Code used by the PDP-1 merely by dropping the eighth bit (parity). A list of characters and their FIO-DEC and Concise Codes can be found in the Appendix.

2. Read Paper Tape, Binary

rpb Address 0002

a. Non-TS: This instruction reads three lines of tape (six channels per line) and assembles the resulting 18-bit word in the Reader Buffer. For a line to be recognized by this instruction, Channel 8 must be punched (lines with Channel 8 not punched will be skipped over). Channel 7 is ignored. The instruction sub 5137, for example, appears on tape and is assembled by rpb as follows:

CHANNEL	8	7	6	5	4	3	2	1
LINE 1	x		x				x	
LINE 2	x		x		x			x
LINE 3	x			x	x	x	x	x

Reader Buffer 100 010 101 001 011 111

NOTE: Vertical dashed line indicates sprocket holes and the symbol "x" indicates holes punched on tape.

NOTE: rpb is initially defined as 730002 in ID symbol table and certainly symbol table.

1) If bit 5 of the rpb instruction is a "1" [730002=rpb], the 18-bit word read from tape is automatically transferred to the IO Register via the Reader Buffer. The program waits until the material has been read and transferred and a completion pulse returned before continuing.
2) If bit 5 of the rpb instruction is a zero [rpb-i=720002], the contents of the Reader Buffer must be transferred to the IO Register by executing a rrb instruction. When the Reader Buffer has information ready to be transferred to the IO Register, status Register bit 1 is set to one. (Sequence break can be used in this situation to perform the read.)

b. TS: The reader must be assigned to the user's console before his program can execute a rpb instruction. (When the reader is not assigned, rpb is an illegal instruction.) This instruction is the same as in NON-TS but the 3 lines read are put into a pseudo reader buffer (prb). The executive routine reader buffer is then filled with succeeding lines of tape. The contents of the prb is then placed in the IO if the instruction was a rpb. Otherwise, on a rpb-i, the material goes into the prb but isn't transferred to the IO until a rrb instruction is executed by the user. Material is taken from the reader buffer in the executive routine on the next rpb instructions. When the buffer is almost empty more tape is read in to refill it.

3. Read Reader Buffer

rrb Address 0030

a. NON-TS: When the rpa or rpb instructions are given with bit 5 a zero (720001=rpa-i or 720002=rpb-i) information read from tape fills the Reader Buffer but is not automatically transferred to the IO Register. To accomplish this transfer, these instructions must be followed by a rrb instructions. In addition the rrb instruction clears Status Register bit 1.

b. TS: Same as above, but the information is stored in the pseudo reader buffer. It is transferred from there to the IO.

B. Perforated Tape Punch

The standard PDP-1 Perforated Tape Punch operates at a speed of 63 lines per second. It can operate in either the alphanumeric or binary mode.

1. Punch Perforated Tape, Alphanumeric

ppa Address 0005

a. NON-TS: This instruction takes information from the IO Register and punches one line of tape in the following format:

IO Bits	10	11	12	13	14	15	16	17
Holes	8	7	6	5	4	3	2	1

NOTE: ppa is initially defined as 730005 in ID symbol table and CERTAINLY symbol table.

1) If bit 5 of the ppa instruction is a "1" [730005=ppa], a completion pulse is generated The program waits until the material has been punched and a completion pulse returned before continuing.

2) If bit 5 is a zero in the ppa instruction [ppa-i=720005], the program is continued and no completion pulse will be given. In this case, the Status bit must be checked for. [Sequence break can be used in this situation to perform the punching].

b. TS: Same as above but transfers one character from IO to the executive routine punch buffer for transfer to the punch when it is ready. If the buffer is full, the user's quantum ends. Bit 5 is completely ignored. Time - about 500 microseconds.

2. Punch Perforated Tape, Binary

ppb Address 0006

One line of tape is punched.as follows:

IO Bits		0	1	2	3	4	5
Holes	8	7	6	5	4	3	2	1

Hole 8 is always punched.

Hole 7 is never punched.

This instruction is like ppa in all other respects.

C. Alphanumeric On-Line Typewriter

The typewriter will operate in the input mode or the output mode.

1. Type Out

tyo Address 0003

a. NON-TS: For each in-out transfer instruction, tyo, one character is typed. The character is specified by the right six bits of the IO Register.
NOTE: tyo is initially defined as 730003 in ID symbol table and CERTAINLY symbol table.

1. If bit 5 of the tyo instruction is a "1" [730003=tyo] a completion pulse is generated. The program waits for the completion pulse before continuing.
2. If bit 5 is a zero in the tyo instruction [tyo-i=720003], the program is continued and no completion pulse will be given. In this case, the status bit must be checked for. (Sequence Break can be used in this situation to perform the typing.)

b. TS: Same as above but places a character from the IO Register into the executive routine user's console buffer to be printed when the typewriter is ready. If his buffer is full, the user's quantum ends. The console is placed in print status by the executive routine and characters are printed from the buffer until it is empty. Then the console returns to type status. Bit 5 is completely ignored. Time: about 500 microseconds.

2. Type In

tyi Address 0004

This operation is completely asynchronous and is therefore handled differently than any of the preceding in-out operations.

a. NON-TS: When a typewriter key is stuck, the 6-bit code for the struck key is placed in the typewriter buffer, program flag 1 is set, and the type-in status bit, bit 3, is set to one. A program designed to accept typed-in data would periodically check program flag 1 or bit 3 of the status register and if found to be set to a one, a tyi instruction could be executed for the information to be transferred to the In-Out Register. This in-out transfer should not use the optional in-out wait. The information contained in the typewriter buffer is then transferred to the right six bits of the In-Out Register. The tyi instruction clears the type-in Status bit.

b. TS: When a typewriter key is struck, the 6-bit code for the struck key is placed in the console's typewriter buffer, the type-in Status, bit 3, is set to one. (Program Flag 1 is not set when a typewriter key is struck.) The tyi instruction alone can replace the listen loop described above to accept typed-in data. Executing this in-out transfer instruction causes a character from the typewriter buffer to be transferred to the right six bits of the In-Out Register. If there are no characters in the buffer, the program is made inactive until a key has been struck.* Programs which perform other operations while waiting for type-in can use a check Status loop or sequence break for detecting the striking of a key. The tyi instruction clears the type-in status bit. Time: about 500 microseconds.

* The executive routine accepts manually typed characters and enters then in the buffer while the console is in type status. If the buffer becomes filled, the type-in light will go out and further characters typed will be lost.

D. Precision CRT Display (type 30)

The sixteen-inch cathode ray tube display is intended to be used as an on-line output device for the PDP-1. It is useful for high speed presentation of graphs, diagrams, drawings, and alphanumerical information. The unit is solid state, self-powered and completely buffered. It has magnetic focus and deflection. The cathode ray tube has a P7 phosphor; thus a point displayed persists for a relatively long time. Some other display characteristics are:

1024 by 1024 addressable locations
fixed origin at center of CRT
Plots 20,000 points per second
Ones complement binary arithmetic
Random point plotting
Accuracy of point ±3 per cent of raster size
Raster size 9.25 by 9.25 inches.

Resolution is such that 512 points along each axis are discernible on the face of the tube.

1. Display One Point on CRT

dpy Address 0X07

a. NON-TS and TS: This instruction clears the light pen status bit and displays one point using the high ten bits (i.e. 0 through 9) of the AC for the X coordinate and the high ten bits of the IO for the Y coordinate. The AC and IO remain unchanged after the dpy instruction. The origin "(o,o)" is at the center of the scope tube and (377400, 400000) is the lower right corner.
The X bits of the instruction control the intensity: 3 is the brightest, 1 normal, 7 barely visible and 4 visible only to photo-multiplier tubes.

If bit 5 of the display instruction is a one, the computer will wait 50 microseconds for the scope to complete before continuing. If bit 5 is off, the program will continue without waiting. The computer will not begin a dpy until the previous one was completed.

dpy is defined in CERTAINLY and ID with bit 5 on, i.e. 730007.

Skip on Display Lever
sdl Address 3477

TS only This instruction skips if and only if the console display lever is depressed. The instruction sdl 1 skips if and only if the display lever is not depressed. When a console's display lever is depressed, the executive routine gives the program running at that console extended running time. This facilitates observing information on the scope.

E. Light Pen (Type 32)

NON-TS and TS: The light pen is designed to be used with the CRT DISPLAY TYPE 30. By "writing" on the face of the CRT, stored or displayed information can be expanded, deleted, or modified. Specifically, the scope completion pulse (about 50 microseconds after the last dpy or dpy - i) interrogates the light pen. If it saw light, the light pen status bit and program flag 3 are set. Note that the light does not have to be from the scope face. A threshold control for the light pen is located on the bottom of the scope tube cabinet.

If a program uses dpy-i instructions and szf 3 to serve the the light pen, there must be at least 45 μ seconds of instructions between the two in order to allow time for Flag 3 to be set by the scope completion pulse.

F. Drum

The PDP-1 drum is a high-speed magnetic drum used for storage It is divided into 22_{10} fields of 4096 words each. Words are transferred between the drum and the core memory under automatic control. The drum runs at 30 revolutions per second: thus, each word on the drum is available once every 33 1/3 milliseconds. When a drum operation has begun, words are transferred at a rate of 8.16 microseconds each. In a single operation, information can be written on the drum, read from the drum or both simultaneously.

1. Drum Initial Address

   dia Address 0060

   a. NON-TS: This instruction causes the $C(IO)_{1-5}$ to be sent to the drum write field buffer. These bits specify which field of the drum will be written on during the next dcc instruction or if $C(IO)_{1-5}$ is zero, that no write operation is to occur. The $C(IO)_{6-17}$ are sent to the drum initial address register to specify the first drum address to be transferred.

   b. TS: The field number employed in a user's program is a pseudo field number unless specified as absolute by the sign bit being on. The pseudo field number is translated into its absolute field number and checked for validity by the executive routine. Time: about 150 microseconds. The IO is stored in ID's "w" register. Checking does not happen until dcc is executed.

2. Drum Break on Address

   dba Address 0061

   a. NON-TS: This instruction causes the $C(IO)_{6-17}$ to be sent to the drum initial address register. When the current drum address becomes equal to the contents of the initial address register, a sequence break request is indicated. Bit 5 of the status word is set by the break, and is cleared by the next dcc instruction.

b. TS: The address given will be advanced by approximately 50 words to allow for processing by the executive routine. Occurrence of the sequence break will interrupt the user's program and turn on the drum status bit. Time: about 200 microseconds.

3. Drum Count and Commence

dcc Address 0062

a. NON-TS: This instruction causes the $C(IO)_{1-5}$ to be sent to the drum read field buffer. These bits specify which field will be read, or if $C(IO)_{1-5}$ is zero, that no read operation is to occur. The $C(AC)_{3-5}$ specify which core will be used for the data transfer; $C(AC)_{6-17}$ specify the first core memory address of the data to be transferred. The $C(IO)_{6-17}$ specify the number of words to be transferred. If the $C(IO)_{6-17}$ is zero, 4096 words are transferred. While the dcc instruction is being executed, the computer stops and the drum system takes full control of the core memory. Successive words are transferred from sequential locations until the operation is complete. If no errors occurred during the drum operation, the instruction following the dcc will be skipped. The C(AC) and the C(IO) are lost during this operation. If both read field and write field are non-zero (both reading and writing operations are specified) the contents of memory are written on the write field; then the read field data are read into memory. The read field must not equal the write field. In order to avoid passing a given drum address, and hence losing 33 milliseconds, the dcc instruction must be given at least 19 microseconds before the drum address reaches the initial address.

b. TS: The field number employed in a user's program is a pseudo field number (unless specified as absolute by the sign bit being on). The pseudo field number is translated into its absolute field number and checked for validity by the executive routine. The C(AC) and C(IO) are preserved. Time: about 300 microseconds plus access and transfer time.

4. Drum Read Address

dra iot 63

a. NON-TS: This instruction causes the current drum address to be read into IO_{6-17}. The parity error flag is read into IO_0: The selection error flag is read into IO_1 and the timing error flag is read into IO_2. Two cycles elapse before this information is placed in the IO.

b. TS: The address read will be about 50 words in advance of the actual position of the durm to allow for processing by the executive routine. Time: about 100 microseconds.

The various error conditions are as follows:

1. Read Error: When a word is written on the drum, its parity is generated and is also written on the drum. Whenever this word is read off the drum, a new parity is generated and is checked against the parity just read. A mismatch sets the read error flip-flop.
2. Selection Error: A five-bit address can specify 32 drum fields out of which 23 are legal and 9 are illegal. Field 0 means no selection, fields 1 through 22 are legal fields and 23 through 32 are illegal fields. A selection error flip-flop will be set if a user selects an illegal field, or if the read field equals the write field.
3. Timing Error: This error monitors the drum clock circuit malfunctions. The timing flip-flop will be set when (a) the time period between consecutive clock pulses is not equal to ~8.2 microseconds, (b) number of clock pulses available on the drum is not equal to 4096, or (c) the reference index pulse is lost.

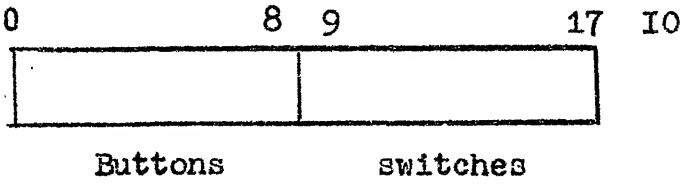
G. Buttons and Switches

Four consoles of nine buttons and nine switches each can be added to the PDP-1 time-sharing system to facilitate the communication between the user and the computer.

1. Read Buttons (10 μsec)

rbt Address x237 where $0 \leq x \leq 3$

TS: This instruction is available only in Time-Sharing. The button console must be assigned to the user's console. [When the button console is not assigned, the rbt clears the IO.] This instruction reads the value of the button and switches into the IO register. The right nine bits of the IO register will contain the state of the switches from left to right (1 meaning on, 0 meaning off); The resulting left nine bits of the IO will contain the buttons from left to right.



The x above corresponds to bits 8-9 of the rbt instruction and indicates one of the four possible sets of buttons and switches (numbered from 0 to 3). The instruction rbt (=720237) reads the lowest numbered console which is assigned to the user; the instruction rbt 400 reads the second lowest numbered console assigned to the user; and so on.

H. Knobs

Four consoles of four analog-devices each can be added to the PDP-1 time sharing system to facilitate the communication between the user and the computer. Two such consoles have been added. Analog voltage from 0 to -10 volts from these devices can be converted into a binary word which and placed in the IO register.

1. Check Knobs (25-30 μsec)

ckn Address xx27 where $0 \leq x \leq 3$

TS: This instruction is available only in Time-Sharing The Analog console must be assigned to the user's console. (When the device is not assigned, the ckn clears the IO.) This instruction reads the value of the selected analog device into the lower 8 bits (bits 10-17) of the IO register. Bits 8-9 of the instruction determine which of the four possible analog consoles numbered from 0 to 3 is selected while bits 10-11 indicate the specific KNOB (numbered 0 to 3 from left to right) on the selected console that is to be read. (Other bits are unused).

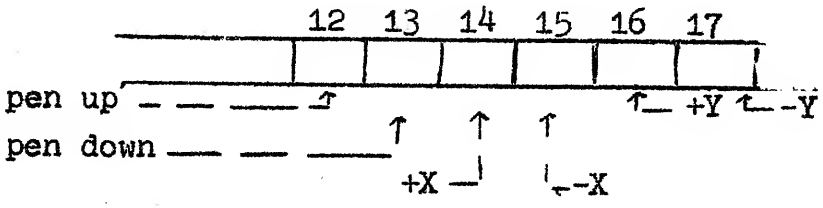
The instructions ckn[=720027), ckn 100, ckn 200, ckn 300, read the lowest numbered console assigned to the user; the instructions ckn 400, ckn 500, ckn 600, ckn 700 read the second lowest numbered console assigned to the user; and so on.

Calcomp Plotter

0.005 inch step size
30 inch by 120 feet maximum paper size
Liquid Ink and ball-point pens
Five ink colors
300 steps per second
7 pen sizes.

The calcomp plotter is activated by the instruction iot 1111. This causes a one step move as specified by the IO register which has the following format:

IO:



It is permissable to move simultaneously in that X and Y directions but not to move in both X or both Y directions. The pen should not be lifted or lowered while it is in motion.

There is no status bit for the plotter so it can not cause sequence breaks or stop the computer. Thus the program is responsible for staying below the 300 step per second rate. (Pen up/down operations take 0.1 second). External level q2 must be assigned.

I. External Register

A 18-bit register has been added to the Time Sharing facilities of the PDP-1 computer to add communication between programs and in-out devices. This External Register can be assigned in two modes, shared or absolute. When it is absolutely assigned, only one user may have it; on shared assignment, it is possible for more than one user to use it.

1. Load External Register from the Accumulator (5 μsec)
   lea Address 4677
   This instruction copies the contents of the Accumulator into the External Register. This instruction is available only in Time Sharing. The external register must be assigned to the user's console. [When the external register is not assigned, the lea does nothing.]
2. Load External Register from the In-Out Register (5 μsec)
   lei Address 4577
   This instruction copies the contents of the External Register into the IO Register. This instruction is available only in Time-Sharing. The external register must be assigned to the user's console. [When the external register is not assigned, the lei does nothing.]
3. Read External Register into the In-Out Register (5 μsec)
   rer Address 4777
   This instruction copies the contents of the External Register into the IO Register. This instruction is available only in Time Sharing.

J/ Sequence Break Mode

The purpose of the Sequence Break Mode (or program interrupt) is to allow concurrent operation of several in-out devices and the main program sequence. It also provides a means of indicating to the computer than an in-out device is ready to accept or furnish data.

a. NON-TS: The standard one channel sequence break mode is the mode available in NON-TS. Interrupt requests can be received from any number of in-out devices. Each such request sets a unique status bit. If the channel is free, the main program sequence is interrupted after completion of the current memory cycle and the C(AC) are automatically stored in memory location 0, the C(PC) in location 1, and the C(IO) in location 2 in memory module zero. The time required to accomplish this is 15 μsec. The C(PC) as stored in location 1 includes the state of the overflow flip-flop in bit 0, the condition of the extend flip-flop in bit 1, and the contents of the Extended Program Counter in bits 3, 4, and 5. The Program Counter is then reset to the address 0003 and the program begins operating in the new sequence. At the beginning of servicing a sequence break, the overflow and extend mode flip-flops are automatically set to zero, and the sequence break hold flip-flop is set to 1. The program beginning at location 0003 is usually designed to inspect the Status bits, through the use of the Check Status Instruction, to determine which in-out device caused the interrupt. A jump to the appropriate in-out subroutine can then be executed. Each such subroutine can be terminated by the following instruction:

```
lac   0000    /to restore the AC
lio   0002    /to restore the IO
jmp i 0001    /to resume the main program
```

If sequence break mode and sequence break hold are on, the jmp i 1 temporarily places the computer in the extend mode so that a 15-bit exit address is obtained, restores the overflow, extend mode, and PC flip-flops to their previous stages (i.e. just prior to the beginning of the sequence break) and turns off sequence break hold, thus allowing the next interrupt request received by the system to be processed. Interrupt requests that occurred while the channel was busy set status bits, and cause interrupts when the channel next becomes free. The read, punch, typewriter, light pen, and drum are attached to the one-channel Sequence Break System and seven status bits are defined (see Check Status Instruction). Three instructions are directly associated with the One-Channel Sequence Break System on the standard PDP-1:

1. Enter Sequence Break Mode
   esm Address 0055
   This instruction turns on the Sequence Break System, allowing automatic interrupts to the main sequence to occur.

2. Leave Sequence Break Mode
   lsm Address 0054
   This instruction turns off the Sequence Break System, Thus preventing interrupts to the main sequence. Should interrupt requests occur while the system is off, the status bit will, nevertheless, continue to be set.

3. Clear Sequence Break System

cbs Address 0056

This instruction clears the sequence break hold flip-flop.

A sixteen channel sequence break system exists in time sharing. This system has sixteen automatic interrupt channels arranged in a priority chain, the lowest channel having the highest priority. A break to a particular sequence can be initiated by the completion of an in-out device, the program, or any external signal. Breaks cannot occur within breaks; they are stored and serviced after the present one is dismissed. If more than one break occurs at the same time, the break with the lowest channel member will be serviced first. When a break occurs, the C(AC) are automatically stored in memory location 0, the C(PC) in location 1, the C(IO) in location 2, all in memory module zero and the number of the channel that caused the break is placed in the AC. The C(PC) as stored in location 1 includes the state of the overflow flip-flop in bit 0 the condition of the extend flip-flop in bit 1 and the contents of the Extended PC in bits 3 through 17. The program counter is then reset to the address 0003 and the program begins operating in the new sequence. For a break to occur, the computer must be in sequence break mode, the sequence break hold flip-flop must be off, and the status bit and the corresponding enable channel associated with this device must be on. At the beginning of servicing a sequence break, the overflow and extend flip-flops are automatically set to zero. The program beginning at location 0003 usually contains a dispatch table since the C(AC) indicates which in-out device caused the interrupt. A jump to the appropriate in-out subroutine can then be executed. Each such subroutine can be terminated by the following instructions:

```
lac     0000    /to restore the AC
lio     0002    /to restore the IO
jmp i   0001    /to resume the main program
```

If sequence break mode and sequence break hold are on, the jmp i 1 temporarily places the computer in the extend mode so that a 15-bit exit address may be obtained, restores the overflow, extend mode and PC flip-flops to their previous stages (i.e., just prior to the beginning of the sequence break), and turns off sequence break hold, thus allowing another interrupt to the processed. The servicing routine should turn off the corresponding status bit. The reader, punch, typewriter, light pen, drum and a set of switches and knobs are each assigned a unique channel in the pseudo 16 channel sequence break system. Provision has also been made for the connection of a user's special external equipment to be two higher and the two lower priority channels. The channels and their corresponding IO devices in the order of their priority are:

Channel	Device
0 1	High priority for user's external equipment
4	buttons
5	light pen
6	type-in
7	type-out
10	punch
11	drum
12	reader
16 17	low priority for user's external equipment

The three instructions associated with the one channel sequence break system of the non-time sharing system are supplemented by three additional instructions in time sharing.

1. Deactivate Sequence Channel N
   dsc N x 100 Address 0050 + (N x100)
   This turns off channel N in the new mode Sequence break system, thus not making it possible for an interrupt to occur on this channel.

2. Activate Sequence Break Channel N
   asc N×100 Address 51+(N×100)
   This instruction turns on channel N in the new mode sequence break system, thus making it possible for an interrupt to occur on this channel.

3. Clear All Channels
   cac Address 53
   This instruction turns off all sixteen channels in the new mode sequence break system, thus not making it possible for any further interrupts to occur until a channel is activated.

K. Memory Module and Memory Extension Control

The standard PDP-1 has a 4096, 18-bit word Memory Module. Additional such modules are connected to the PDP-1 for expanding memory capacity. A memory extension control is needed to expand memory in increments of 4096-word modules beyond the standard 4096 words. This control provides a single-level, indirect address mode called "extend", in addition to the normal multiple-level, indirect address mode of the standard PDP-1. During the operation of a program the extend or normal mode can be selected as required through the use of two instructions:

1. Enter Extend Mode (5 μsec)

eem 770046

NON TS and TS: This instruction places the computer in the single-level, indirect address mode called "extend". In this mode, all memory reference instruction that are indirectly addressed refer to the location of a word which is taken as a 15-bit effective address. This address is contained in bits 3 through 17 of the specified word. The Program Counter (PC) and the Memory Address Register (MA) both become 15-bit registers. As in multiple-level indirect address mode, the instructions jsp, jda, cal, and jdp supply the AC or memory with the state of the overflow flip-flop in bit zero, the state of the indirect address mode (extend=1, normal=0) in bit 1, and the contents of the extended Program Counter in bit 3 through 17. Instructions not indirectly addressed are executed as in the standard PDP-1, but refer to the 4096 words in the memory module designated by the program counter extension, PC bits 3 through 5. Only bits 6 through 17 of the extended Program Counter act as a counter. Therefore, unless a transfer of control is indicated, an instruction in location 7777 is followed by the instruction in location 0000 of the same memory module, as specified by PC bits 3 through 5. In the extend mode, the cal instruction uses memory location 0100 and 0101 in memory module zero.

2. Leave Extend Mode (5 μsec)

lem 770047

NON TS and TS: This instruction places the computer in the multiple-level, indirect address mode called "normal". In this mode, the PDP-1 operates as usual and all addressing refers to the 4096 words in the memory module designated by the program counter extension, PC bits 3 through 5. As in the extend mode, the instructions jsp, jda, cal, and jdp supply the AC with the contents of the overflow, indirect address mode, and PC flip-flops. In the normal mode, the cal instruction uses memory locations 0100 and 0101 in memory module zero.

L. Check Status

cks Address 0033

This instruction checks the status of various in-out devices and sets IO Bits 0 through 6 for subsequent program interrogation as follows:

a. NON-TS:

IO Bit Position	Status Register Definitions
0	Set to 1 if a light-pulse strikes pen 50 μs after the start of a dpy. Set to 0 at the start of each dpy instruction.
1	Set to 1 when Paper Tape Reader Buffer has information ready to be transferred to IO Register. Set to 0 by the reader return pulse or by the rrb instruction.
2	Set to 1 when typewriter is free to receive a tyo instruction Set to 0 at the start of each tyo instruction.
3.	Set to 1 when typewriter key is struck. Set to 0 by the completion of tyi instruction.
4.	Set to 1 when paper tape punch is free to receive a ppa or ppb instruction. Set to 0 at the start of each ppa or ppb instruction.
5.	Set to 1 when drum address equals address specified by dba instruction Set to 0 by the dcc instruction
6	Set to 1 on entering the Sequence Break Mode. Set to 0 on leaving the Sequence Break Mode.

b. TS:

IO Bit Position	Status Register Definitions
0	Set to 1 if a light-pulse strikes the pen. 50μs after the start of a dpy instruction. Set to 0 at the start of each dpy instruction.
1	Set at 1 when information is in the pseudo reader buffer which can be read into the IO by rpa, rpb, or rrb. 0 if not.
2	Set to 1 when executive routine can accept a character via a tyo trap. Set to 0 when it can't.
3	Set to 1 when executive routine can supply a character via a tyi trap. Set to 0 when it can't.
4	Set to 1 when executive routine can accept a character via a ppa or ppb trap. Set to 0 when it can't
5	Set to 1 when drum break return occurs. Never set to 0.
6	Set to 1 on entering the Sequence Break Mode. Set to 0 on leaving the Sequence Break Mode.

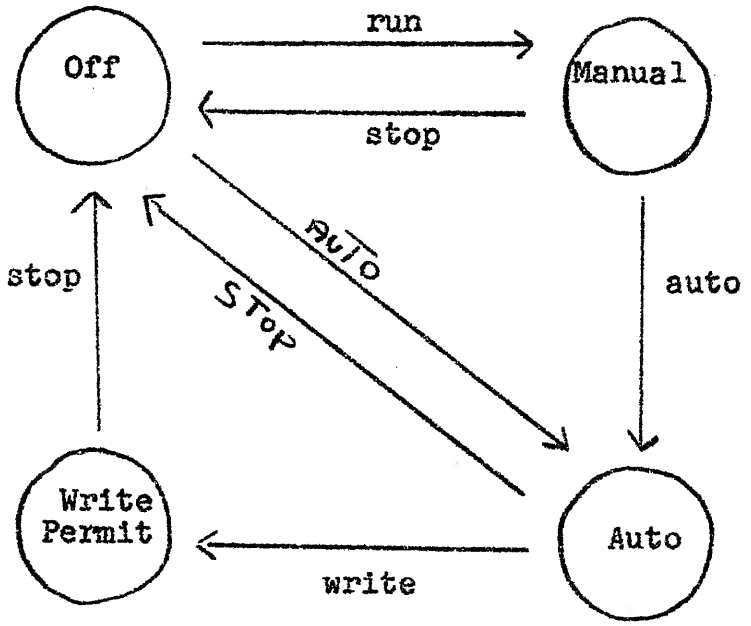
Additional Status bits exist in time sharing.
They cause sequence breaks if the corresponding channel is enabled, but are not read by the cks instruction.

clock (channel 1)	Set to 1 if external level 4 is assigned and a clock pulse occurs. Set to 0 by iot 610 if level 4 is assigned.
buttons	Set to 1 when a change occurs in the state of an assigned button console. Set to 0 by rbt instruction appropriate to button console on which the change occurred.

M. MICROTAPE

Mechanical operation of the tape drives

The following state diagram shows how the drives are controlled.



When a drive is off, the motors are not operating, and a tape may be mounted or removed. When in manual mode, the motors are under control of the "fwd" and "rev" buttons. When in automatic or write permit modes, the motors are under control of the computer and the tape may be read or written under program control.

To mount a tape, press it firmly onto the left hub, draw the tape over the head and onto the empty reel on the right hub, and wind it onto that reel one or two turns by hand. Press the "run" button to place the tape in manual status, and run the tape forward for a few seconds. Then press the "auto" button, followed by the "write" button if desired.

To remove a tape, press the "stop" button, followed by the "run" button. Move the tape in reverse by means of the "rev" button until it comes completely off the take-up reel. Press "stop", stop the coasting tape by hand, and remove it.

Assignment and dismissal of microtapes

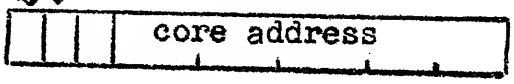
The arq instruction with 4423 (mt") in the accumulator assigns the absolute tape unit in the instruction part of the IO to the pseudo number (0 to 17) in the low 6 bits of the IO. If the low six IO bits contain zero, the tape will be assigned to the first free pseudo number. Pseudo microtape numbers are stored in the drum field translation table, and it is not possible to assign pseudo microtape n and pseudo drum field n+1. If the assignment is successful, the arq skips and places the pseudo number in the low IO. The arq instruction with 773354 (-mt") in the AC dismisses the pseudo microtape number given in the low 6 bits of the IO. It never skips, whether the tape unit was assigned or not. An attempt to assign a micro tape unit will not skip if a microtape is already assigned to the same pseudo number.

Operation of microtapes

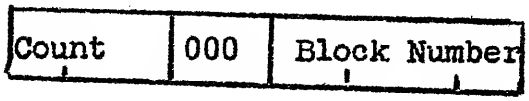
Each reel of tape has 1000 blocks of 400 words each.

The instruction ivk n, where ivk has op code 74 and n is a pseudo microtape number, operates that tape unit. The core address of the first word of the first block is in the AC bits 3-17. The transfer may be in any core. Bits 0 and 1 of the AC are decoded as follows

00 - read
10 - write
x1 - rewind



AC



IO

The right half of the IO gives the block number of the first block to be transferred. Bits 0 to 5 give the number of blocks. This number should not be greater than 40. If it is zero, one block will be transferred. Consecutive blocks are transferred to consecutive areas in core. Block 0 is considered to follow block 777. No block may be transferred partly into one core and partly into another, but different blocks may be transferred to different cores in the same operation. If all blocks are transferred successfully, the instruction skips, leaving the address of the last word transferred+1 in the AC with bits 0 and 1 unchanged, the number of the last block transferred+1 in the right half of the IO, and zero in bits 0 to 5 of the IO. If an error occurs on any block, the instruction does not skip, the right half of the IO contains the number of the block in which the error occurred, and IO bits 0 to 5 contain the number of blocks remaining, including the one that was in error. Furthermore, an error code is placed in the AC.

0 - tape unit is not in automatic status
1 - block cannot be found (probably bad tape)
2 - illegal core address
3 - check sum error (the data transfer took place anyway)
4 - mark track error (probably bad tape)
5 - data channel error (serious hardware malfunction)
6 - no write permit

IV. Instructions for Communication Between User and the Time-Sharing System

The following instructions were added to the time-sharing PDP-1 system so that the user could communicate to the system itself. These are available only in time-sharing mode and are transparent to the accumulator and the in-out register unless otherwise stated. All these instructions trap to the executive routine to the user's program upon completion of the instruction.

1. Administrative Request

a. arq Address 2277

This instruction is used in the time sharing system to assign and deassign in-out equipment and additional drum fields to users. (If the device is not assigned to the user, the instructions corresponding to that device are treated as illegal.) The particular assignment or deassignment requested by this instruction is indicated by mnemonic codes. Concise codes for thes mnemonics are placed in the AC and any additional information necessary for the request is placed in the IO before the arq instruction is executed. On the next two pages is the table of possible requests.

If the assignment or deassignment of fields is successful, the instruction following the arq will be skipped. For other assignments and deassignments, the instruction following the arq will be skipped only on successful assignment. An assignment will be successful if the field (s) or device requested is not already assigned or if the the assignment is already in effect.

The instruction "arq i" skips when and only when the corresponding arq does not.

MNEMONIC WHOSE CONCISE CODE IS CONTENTS OF AC	CONTENTS OF IO	REQUEST
-r	---	dismiss reader
r	---	assign reader
-p	---	dismiss reader
p	---	assign punch
-x	---	dismiss external register
ax	---	assign external register absolutely.
sx	---	assign external register in shared mode.
-c1	---	dismiss core 1
c1	---	assign core 1: this will be unsuccessful when no drum fields are available since one drum field is used for each user that has core 1 assigned.
k	M	assign or deassign analog-to-digital consoles. M is a 4-bit mask for consoles to be assigned (or left assigned, in the case of deassignment) to the user.
b	M	Assign or deassign button consoles. M is a 4-bit mask for the consoles to be assigned (or left assigned, in the case of deassignment) to the user.
q1	---	assign external level 1
q2	---	" " " 2
q3	---	" " " 3
q4	---	" " " 4
q5	---	" " " 5
q6	---	" " " 6
q7	---	" " " 7
-q1 thru -q7		deassign external level 1 thru 7 respectively.

MNEMONIC WHOSE CONCISE CODE IS CONTENTS OF AC	CONTENTS OF IO	REQUEST
-q	---	deassign all external levels
-f	---	dismiss all fields
f	N×10000	assign N additional fields in absolute mode to lowest available pseudo field numbers.
-1f	---	dismiss one field
1f	---	assign, in absolute mode, one field- returns with pseudo field just assigned in high part of AC
af	A×10000+P	assign in absolute mode, absolute field A (or the first available field if A=0) to pseudo field P (or the first available pseudo field if A=0). Return with pseudo field in high part of AC.
-af	A×10000+P	case 1: P=x, A=0. deassign pseudo field x and the absolute field assigned to it. case 2: P=0, A=y. deassign absolute field y and the pseudo field assigned to it. case 3: P=x, A=y. deassign pseudo field X and the corresponding absolute field y. If x does not correspond, no deassignment is done and the request is unsuccessful. case 4: P=0, A=0. No deassignment is done, but the request is successful.

MNEMONIC WHOSE CONCISE CODE IS CONTENTS OF AC	CONTENTS OF IO	REQUEST
tf	---	translate pseudo field P and returns with its absolute field number in high 6-bits of AC.
sf	A×10000+P	assign, in shared mode, absolute field A (or the first available field if A=0) to pseudo field P (or the first available pseudo field if P=0). Returns with pseudo field in high 6-bits of AC.
-sf	---	same as -af.
xf	P_1×10000+P_2	exchange pseudo fields P_1 and P_2 so that they correspond to the other's absolute field. This arq does not skip.
nf	---	counts the number of fields assigned (shared or absolutely) to this console and returns with this number in high part of AC.
mb	---	Returns memory bound in AC, i.e. 1+the highest legal address the user can reference.
mt	A×10000+P	Assign absolute DECtape unit A (0 or 1) to pseudo capability index P (1 to 17, or first free index if P=0).
-mt	P	dismiss pseudo DECtape number P.

2. Dismiss
dsm Address 2377
This instruction is available only in time-sharing so that a user can dismiss his program and can return control to ID. ID is brought back into control and types a carriage return.

3. Breakpoint
bpt 770044
This instruction is available only in TS. It is provided so that ID can insert it into the user's program where breakpoints are requested for debugging. When the user transfers control from ID to his program, a bpt instruction replaces the instruction previously at the breakpoint location. When this instruction is encountered, ID types the address at which the trap occurred followed by a ")" and a tab. Then the previous contents of that register are typed out and the current location pointer is set to that address. If a bpt is encountered at a location where a breakpoint was not assigned to the user through ID, then ID interprets the instruction as illegal.
NOTE: Do not attempt to break at a program-modified or program-read instruction or at an instruction in the middle of a chain of indirect addressing.

4. Wait
wat Address 2477
This instruction is available only in time-sharing and is provided so that the user may deactivate his program and wait until a sequence break occurs. When the break occurs, the contents of location 1 is the location of the wat instruction.

V. ADDRESS MODES AND MICROPROGRAM CLASS

A. Address MODES

The index register is an active register which is particularly useful for modifying operand addresses. Its name derives from its use in indexing through arrays since its contents (the index) can be added to the address field of an instruction (array base address) to determine the effective address for the desired operand. In dealing with linked data structures it is often convenient to load a "pointer" into the index register and to have a "displacement" quantity in the address field of the instruction. Thus the effective address becomes the pointer value offset by the displacement.

On the PDP-1 the index register (XR) is an 18-bit register Effective address calculation is performed by one's complement addition of the XR and the address field of the instruction. The result is a 15-bit extended address which is relative to the current core if the machine is not in extend mode. If the machine is in extend mode, the 15-bit index sum is absolute. The indexing operation does not require any time in addition to the instruction time.

Since there are no spare bits available in the op code to designate the indexing operation the "i" bit (bit 5) is used to designate either indirection or indexing, depending upon the state of the machine as set by mode instructions. The machine remains in any given mode until another mode setting instruction is executed.
Mode setting instructions are:

1. Normal Address Mode
   nam op code 770052
   Memory referencing instructions containing an "i" bit will be indirected.
2. Index Address Mode
   iam op code 770054
   Memory referencing instructions containing an "i" bit will be indexed.

3. Base Address Mode

bam op code 770053

Memory referencing instructions are always indexed. If the "i" bit is on, the instruction is also indirected, and indexing occurs first.

4. Defer Address Mode

dam op code 770055

Memory referencing instructions are always indirected. If the "i" bit is on, the instruction is also indexed, and indexing occurs first.

5. Alternate Address Mode

aam op code 770056

The instruction following the aam is executed in an address mode different from the nominal address mode as follows:

nam ⟷ bam

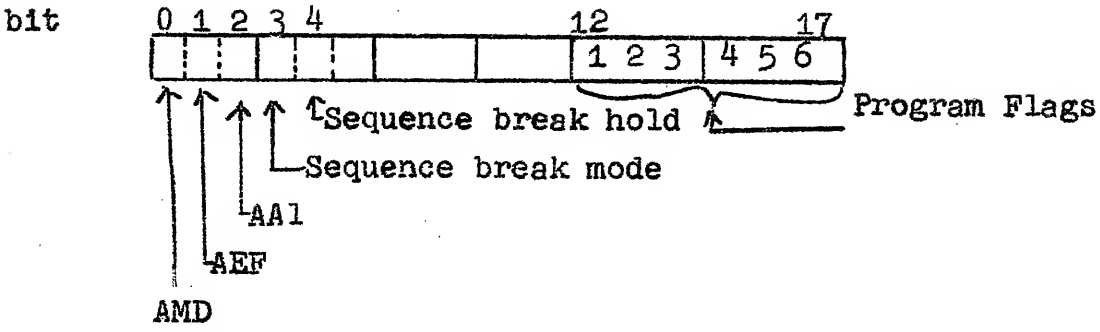
iam ⟷ dam

If xct follows aam, the xct itself will be executed in the alternated mode, but the instruction which the xct fetches will be executed in the nominal mode.

The nominal mode for the instructions jmp, jsp, jdp, jda is always taken to be normal. However, aam will alternate the address mode. The effective address of cal is always 0100 in core 0.

B. FLAG WORD

There is an 18-bit flag word, the register "F" of ID, containing the following:



AMD and AEF determine the nominal address mode:

AMD	AEF	MODE
0	0	nam
0	1	bam
1	0	iam
1	1	dam

AAL is complemented by the aam instruction. If AAL is on, the next instruction is executed in alternated address mode unless the instruction is in an xct chain and not the first xct of the chain, and then AA1 is turned off. If AAL is on, sequence breaks will not occur.

Read Program Flags

rpf op code 770050

This instruction reads bits 0-2, 12-17 of the flag word into the IO and clears bits 3-11 of the ID.

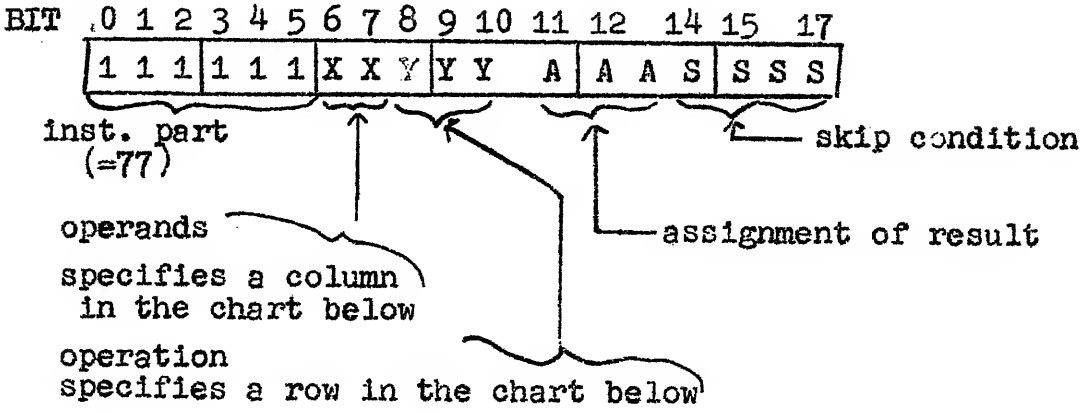
Load Program Flags

lpf op code 770051

This instruction loads bits 0-2, 12-17 of the IO into the flag word.

C. MICRO-PROGRAM CLASS

The micro-program instruction class has the following instruction format.



symbolic specification for Micro-program operations

A =AC, I = IO, X = XR

YYY \ XX				OPERATION (result)
000		TI	TA	TX	T (True, test, or transfer)
001		CI	CA	CX	C (Complement (negative))
010		I (A→I)	A (X→A)	A (X→I)	exchange (see explanation below)
011		AMI	XMA	XMI	M (arithmetic Minus)
100	Z (zero)	AVI	XVA	XVI	V (inclusive OR)
101	SA (AC) +1	AΛI	XΛA	XΛI	Λ (Bitwise AND)
110	SI (IO) +1	A~I	X~A	X~I	~ (exclusive OR)
111	SX (XR) +1	A+I	X+A	X+I	+ (arithmetic Plus)

S (Step, ie., add one)

The functions of the result assignment and skip condition fields are as follows:

AAA	Bit 11=1 will put the result in the AC		(A)
	Bit 12=1 " " " " " " IO		(I)
	Bit 13=1 " " " " " " XR		(X)
SSSS:	Bit 14=1 will cause a skip if the result is >+0		(>)
	Bit 15=1 " " " " " " " <-0		(<)
	Bit 16=1 " " " " " " " = +0		(P)
	Bit 17=1 " " " " " " " = -0		(M)

The execution of micro-program instruction computes the result specified by the XX and YYY bits (i.e. from Table on previous page), puts this result in the specified register (s) and skips if any of the specified conditions are true. Note that skip condition is evaluated on the result of the micro-program, not the final contents of any given register, and the result need not be assigned to any register. Thus, it is possible to test the sum of the AC and IO to see if it is equal to -0 or greater than +0 without destroying the contents of these registers. The instruction to do this is 773611 or, symbolicly, A+IM>. All opr i instructions take 5 seconds.

The "T" operation clears the transmitted register after the transfer. This may be prevented by assigning the result back to the transmitted register. See the examples.

The exchange operation is a special case. The result I (A→I) means that the result of the function is the old IO register, but in addition the accumulator is placed in the IO. This secondary assignment is done before the assignment given by the AAA bits in the instruction. Thus, the instruction A→I (772400) will move the AC to the IO, A→IA (772500) will swap the AC and IO, and A→II (772440) does nothing (because the primary assignment is to the IO and the result is the old IO).

If the result of an add or subtract micro-program instruction is zero, the result is +0 except in two cases:

(-0) + (-0) = (-0)
(-0) - (+0) = (-0)

The step instructions steps (-1) to (+0).
Micro-program instructions never turn on the overflow flip-flop.

USACE

The assembler considers certain symbols consisting of capital letters as micro-program instructions. The entire instruction must be in upper case, and may appear in any expression, e.g., storage word, constant, etc. When typing into ID the instruction must be preceded by a single quote (').

Micro-programs are specified by concatenating three "fields" - the result field, the assignment field, and the skip field. The characters in all of these must be in upper case and there must be no separator between the fields.

<result field><assignment field><skip field>

<result field> must be one of the twenty-eight results given in the Table on a previous page.

<assignment field> may be null (no characters) or any combination of A, I, and X to specify in which registers the result will be placed.

<skip field> may be null or contain any combination of >, <, P, M, |, _, and =.

> means skip if result is greater than +0
P means skip if result is equal to +0
M means skip if result is equal to -0
< means skip if result is less than -0
| means complement the specified skip conditions
= and _ means skip if result is either +0 or -0

Since the octal representation of a micro-program instruction is computed by exclusive - OR' ing all of the specifications within each field, redundant specifications may lead to unexpected results.

For example TAIII is the same as TAI: and TAI>> is the same as TAI. An error in syntax results in a uer error message from the assembler.

MICRO-PROGRAM INSTRUCTION CLASS

Sample Micro-program instructions

symbolic	octal	action
A+I	773600	computes sum of AC and IO and does nothing at all with it.
A+IA	773700	the sum of the AC and IO is put into the AC.
A+IAIX	773760	the sum of the AC and IO is put into the AC, IO, and XR.
A+I<P	773606	skips if the sum of the AC and IO is less than -0 or equal to +0.
A+I<P|	773611	skip if the sum of the AC and IO is not less than -0 or equal to +0.
A+IX≥	773633	the sum of the AC and IO is put into the XR. If the sum was -0, +0, or greater than +0, the instruction, will skip.
X→IXA|	776437	exchanges the IO and XR. The "old" IO is also put into the AC. This instruction skips always.
ZAIX	771160	the AC, IO, and XR are cleared.
SA>	771210	skip if the AC plus one is greater than +0.
SAA?	771302	add one (step) to the AC and skip if it is +0.
TXM	776001	skip if the XR is -0 and clear the XR.
CXP	776202	skip if the XR is -0.
TAXI	774060	transfer the contents of the AC into the XR and IO, the AC is cleared.
TAAXI	774160	same as TAXI, but the AC is not cleared.

EXAMPLES:

Program which stores zeroes in all of array2.

```
n=100

dimension array2 (n)

        iam                 /enter indexing mode
        lxr (-n             /initial value into XR
        dzm i array2+n      /store a zero into the array
        SXXP                /step xr and skip if it is +0
        jmp .-2             /continue loop
          .
          .                 /that does it.
```

To restore address mode after sequence break service,

assuming sb service is in nam or iam.

```
          .
          .
        lxr   XR            /saved XR
        lac   o             /saved AC
        lio   flg           /saved flag word
        lpf
        ril   1
        spi
        eam                 /undo  bam or dam
        lio   2             /saved IO
        jmp i 1             /dismiss
```

ADDITIONAL NOTES

1. Due to hardware changes which are in progress, the time estimates given in this list may be wrong. In general it is very difficult to say how long a sequence of instructions will take to execute--in time-sharing, it is impossible.

2. No attempt has been made to descirbe the *ivk* (740000) instruction or the I/O Function bus because most I/O devices are still operated by *iot* instructions.

3. Defer address mode (see *dam*) will be changed in the future.